FAT-Pointer based range TLB

Akilan Selvacoumar[1], Robert Stewart[1], Hans-Wolfgang Loidl[1], and Ryad Soobhany[1]

[1]Heriot-Watt University, UK

Abstract

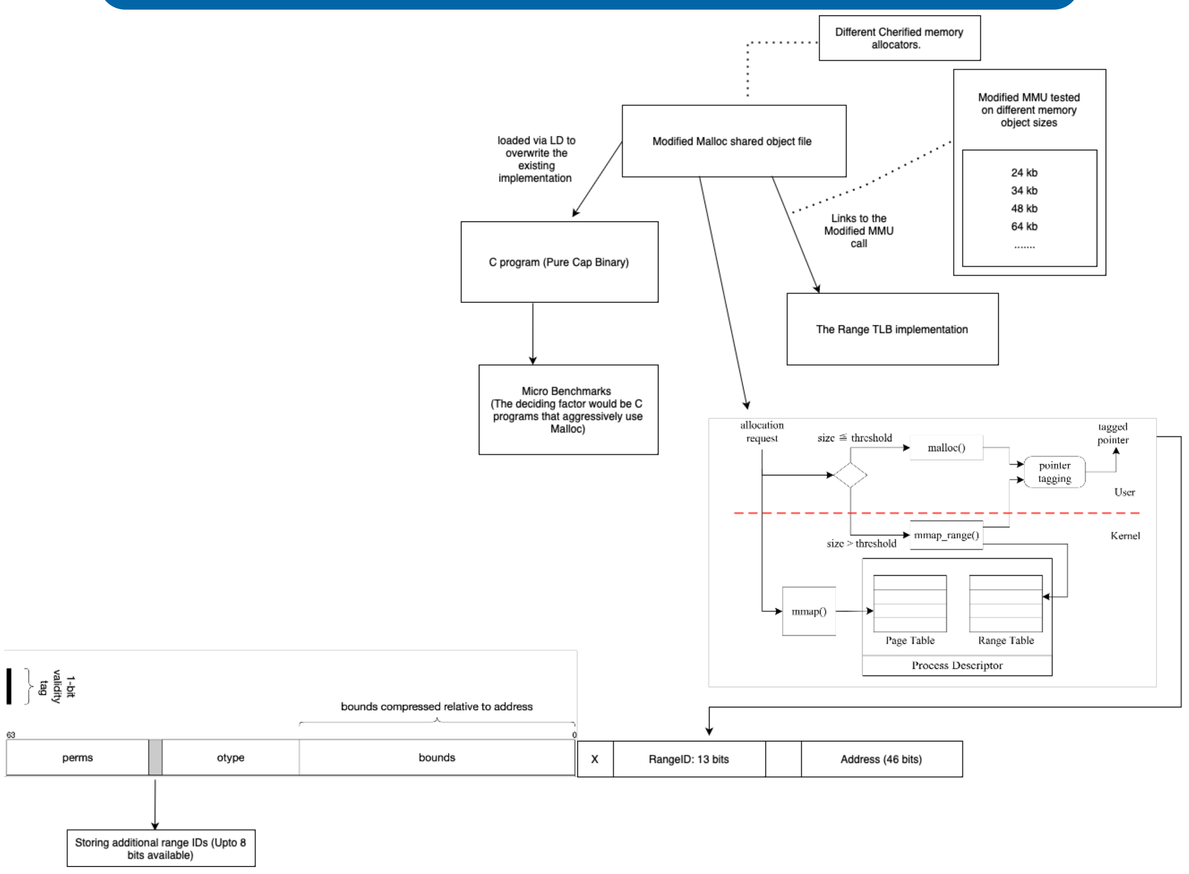
The CHERI architecture is designed to provide security guranetees, This has had performance implications such as more L1 TLB misses on purecap mode rather than running an application as a 64 bit program. The following expirement proposes on adding range TLB which is stored on the page table to store TLB translation entries of large memory objects (This would inturn lookup TLB entries in the range table rather than looking up the L1 TLB entry). During this process the range ID would be added to the pointer which allows looking up the TLB entries at an earlier stage during the pipeline. The FAT pointer structure of CHERI allows to encode custom bits of meta-data which in this case can be used to store the ID of the range TLB entry.

Research Questions and Goals

Research Questions:

- (RQ1) Is there an optimization that explicitily uses FAT pointers to reduce the number of TLB misses to reduce the CPU STALL Time ?
- (RQ2) Can the number of L1 DTLB misses be minised by offloading translation lookups to the newly introduced range TLB ?

Expirement Proposal



Exploiting CHERI pointers

References